Information and Computation 207 (2009)1340–1349

Contents lists available at ScienceDirect

Information and Computation

journal homepage: www.elsevier.com/locate/ic

Variants of codes and indecomposable languages

Arto Salomaa [a,*], Kai Salomaa [b], Sheng Yu [c]

[a] Turku Centre for Computer Science, Joukahaisenkatu 3-5 B, 20520 Turku, Finland
[b] Department of Computing and Information Science, Queen's University, Kingston, Ont., Canada K7L 3N6
[c] Department of Computer Science, The University of Western Ontario, London, Ont., Canada N6A 5B7

ARTICLE INFO

Article history:
Received 4 June 2008
Available online 27 March 2009

Keywords:
Length code
Catenation of languages
Language
Decomposition
Prime decomposition
Star language

ABSTRACT

We continue the investigation of representing a language as a catenation of languages, each of which cannot be further decomposed in a nontrivial fashion. We study such prime decompositions, both finite and infinite ones. The notion of a length code, an extension of the notion of a code leads to general results concerning decompositions of star languages. Special emphasis is on the decomposition of regular languages. Also some open problems are mentioned.

© 2009 Elsevier Inc. All rights reserved.

1. Introduction

Products or catenations of languages, viewed as subsets of the free monoid, are needed in many applications. However, because of the noncommutativity, many phenomena are not yet properly understood. For instance, although the condition for the commutation of two words can be explicitly stated, the equation $L_1L_2 = L_2L_1$ for languages presents various difficulties. (See [13,9] and their references.)

This paper investigates products of languages where the individual factors L cannot be decomposed further in a nontrivial way, that is, presented in the form $L = L_1L_2$, where neither of the languages L_1 and L_2 consists of the empty word alone. The initial work on such "prime decompositions" [13,11] concentrated mainly on finite languages. Then a prime decomposition can always be found, although it is not necessarily unique. It is decidable whether or not a regular language is prime but the complexity of this problem is not known.

Various cases were considered in [5] where a language has a prime decomposition consisting of infinitely many factors but none with a finite number of factors. This paper continues the investigation of such finitary and infinitary prime decompositions. A generalization of the notion of a code, a *length code*, will be a useful tool in this investigation. In this paper, we are mainly concerned with regular languages. As will be seen below, it makes a big difference whether or not the prime factors of a regular language are also required to be regular.

There has been much work recently concerning orthogonal (unambiguous) catenation. (See, for instance [3].) Studies similar to those in this paper could be carried out also with respect to orthogonal catenation, instead of ordinary catenation. For instance, it is an open problem whether for a given regular language L we can effectively decide whether L can be written in a nontrivial way as an orthogonal catenation of two languages. The question is open also if we additionally require that the component languages be regular [3]. The corresponding questions are undecidable for context-free languages.

* Corresponding author.
E-mail addresses: asalomaa@utu.fi (A. Salomaa), ksalomaa@cs.queensu.ca (K. Salomaa), syu@csd.uwo.ca (S. Yu).

0890-5401/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.ic.2009.03.004

A brief outline about the contents of this paper follows. The next section discusses various possibilities of defining a prime decomposition and presents examples of the different cases. More explicit comparisons are made in Section 3. The notion of a *length code* is introduced and its basic properties are discussed in Section 4, and its connection with prime decompositions of star languages is shown. Techniques for obtaining prime decompositions, beyond those using length codes, are presented in Section 5. In case of regular languages, the techniques lead to nonregular factors. Possibilities of actually getting regular factors are discussed in Section 6.

2. Different types of prime factorizations

We assume that the reader is familiar with the basics of formal languages. Whenever necessary [12], may be consulted. As customary, we use small letters from the beginning of the English alphabet a, b, c, d, possibly with indices, to denote letters of our formal alphabet Σ. Words are usually denoted by small letters from the end of the English alphabet. The empty word is denoted by ε. Following the regular expression notation, we sometimes denote the union by "+" and singleton sets $\{\alpha\}$ simply by α. Thus, $\varepsilon + ab$ stands for the set $\{\varepsilon\} \cup \{ab\}$. The following definition [11] contains the basic notions of this paper.

Definition 1. A nonempty language L has a *nontrivial decomposition* if, for some $L_1 \neq \{\varepsilon\}$ and $L_2 \neq \{\varepsilon\}$, we have $L = L_1L_2$. A nonempty language $L \neq \{\varepsilon\}$ having no nontrivial decomposition is *prime*. A language L has a *prime factorization* (or a *prime decomposition*) if

$$L = L_1 \cdots L_m, \quad m \geq 1,$$

where each of the languages L_i, $1 \leq i \leq m$, is prime.

Observe that by a "prime factorization" without further specifications we mean a *finite* factorization. Another definition is given below for infinitary prime factorizations.

Products of subsets of the free monoid are not yet very well understood. One can also visualize different ways of defining a "prime" language. For instance, in [4] a language L is termed *indecomposable* if the equation $L = L_1L_2$ implies that either $L = L_1$ or $L = L_2$. It is clear that if a language is prime in the sense of Definition 1, then it is also indecomposable. The languages $\emptyset$ and $\{\varepsilon\}$ are, by definition, not prime although they are indecomposable. Below we show that, besides these trivial cases, the notions of primality and indecomposability coincide.

Theorem 1. *Let L be a language distinct from $\emptyset$ and $\{\varepsilon\}$. If L is indecomposable, then L is prime.*

Proof. Let $L \neq \emptyset, \{\varepsilon\}$ be an arbitrary indecomposable language.

For the sake of contradiction we assume that L is not prime, that is, we can write

$$L = L_1L_2, \tag{1}$$

where $L_i \neq \{\varepsilon\}$, $i = 1, 2$. Below by considering three different possibilities we derive a contradiction.

(a) Assume that $L_1 = L_2 = L$. From (1) we know that L has to be infinite and, furthermore, $\varepsilon \in L$ because otherwise LL cannot contain any word of L of minimal length.
Choose disjoint sets $M_i \neq \emptyset, \{\varepsilon\}$, $i = 1, 2$, such that $L = M_1 + M_2$. Now

$$L = (\{\varepsilon\} + M_1)(\{\varepsilon\} + M_2). \tag{2}$$

Above the inclusion from right to left follows from the fact that $\{\varepsilon\} + M_i \subseteq L$, $i = 1, 2$, and $L = LL$. On the other hand, the right side of (2) contains $M_1 + M_2$ which gives the inclusion in the other direction. Since $\{\varepsilon\} + M_i \neq L$, $i = 1, 2$, (2) contradicts the assumption that L is indecomposable.

(b) Assume that $L_1 = L$ and $L_2 \neq L$. Since $L_1, L_2 \neq \{\varepsilon\}$ we can choose nonempty words $u_i \in L_i$, $i = 1, 2$. We observe that $u_1u_2 \in L_1L_2 = L_1$ and define $L_1' = L_1 - \{u_1u_2\}$. We claim that

$$L = L_1'L_2. \tag{3}$$

Inclusion from right to left follows from (1) and the fact that $L_1' \subset L_1$. To show the inclusion from left to right consider an arbitrary $w \in L$. First assume that $w \neq u_1u_2$. The equation $L = L_1L_2$ implies that $\varepsilon \in L_2$ because otherwise L_1L_2 could not contain the words of L $(= L_1)$ having minimal length. Now w can be written as the catenation of words $w \in L_1'$ $(= L - \{u_1u_2\})$ and $\varepsilon \in L_2$. Finally, the word $w = u_1u_2$ is the catenation of $u_1 \in L_1'$ and $u_2 \in L_2$.
Since $L_1' \neq L$ and $L_2 \neq L$, (3) contradicts the assumption that L is indecomposable.

(c) Finally, the case where $L_1 \neq L$ and $L_2 = L$ is symmetric to (b) above. □

We can consider the following weaker definition of indecomposability for regular languages. We say that a regular language L is *regularly indecomposable* if, for any factorization of L into regular components L_1L_2, one of the equations $L = L_i$ has to hold, $i \in \{1, 2\}$. That is, L being regularly indecomposable leaves open the possibility that L could have an arbitrary decomposition in terms of nonregular languages.

As a corollary of the proof of Theorem 1 we get the following slightly stronger statement for the correspondence of regular indecomposability and primality.

Corollary 1. *Let L be a regular language distinct from $\emptyset$ and $\{\varepsilon\}$. If L is regularly indecomposable, then L is prime.*

Proof. As in the proof of Theorem 1 we assume that L is not prime and write $L = L_1L_2$, $L_i \neq \{\varepsilon\}$, $i = 1, 2$. Since L is regular, from [11,13] we know that there exist regular languages $R_i \supseteq L_i$, $i = 1, 2$, such that $L = R_1R_2$. After this the proof proceeds exactly as before. Note, in particular, that in case (a) (corresponding to the possibility $R_1 = R_2 = L$) the languages M_1 and M_2 can obviously be chosen to be regular which means that we get a contradiction with regular indecomposability of L. □

It is obvious that every finite language has a prime factorization. It is not necessarily unique, for instance,

$$(\varepsilon + a^2 + a^3)(\varepsilon + a^3 + a^4) = (\varepsilon + a^2)(\varepsilon + a^3 + a^4 + a^5),$$

where it is easy to verify that all four factors listed are indeed prime. A prefix-free regular language has a *unique* prime factorization if it is additionally required that the factors are regular prefix-free languages [2,7] but infix-free regular languages do not possess the analogous property [6]. Decompositions of factorial languages, that is, languages closed under the subword operation are investigated in [1,4].

The notion of a *strongly prime decomposable* language was introduced in [5]. A language L is strongly prime decomposable if, for some integer t, any decomposition of L contains at most t nontrivial factors. When L is strongly prime decomposable, any way of iteratively decomposing L has to stop after a finite number of steps, i.e., the refinement of any decomposition results in a prime decomposition in a finite number of steps. In this case we say also that L has a *strong finitary prime decomposition*.

The language Σ^* possesses many nontrivial decompositions. Some of them are prime factorizations, as shown in [5]. Indeed, consider any nonempty language $H \subseteq \Sigma^n$, $n \geq 1$. (Thus, possibly $H = \Sigma$.) Then

$$H^* = (\varepsilon + H)\left(\varepsilon + H(H^2)^*\right),$$

where both factors on the right side are prime.

It is also shown in [5] that the language

$$H_d = \varepsilon + \left\{a^{i_1}b^{i_1}a^{i_2}b^{i_2}\cdots a^{i_k}b^{i_k} \mid k \geq 1,\ 1 \leq i_1 < i_2 < \cdots < i_k\right\}$$

does not have a prime factorization. (The language H_d is not context-free but its complement is context-free.) This reflects the fact that Definition 1 requires the prime factorization to be finitary. If this requirement is relaxed, we can write

$$H_d = \prod_{i=1}^{\infty}\left(\varepsilon + a^ib^i\right),$$

where each factor is prime. Moreover, this infinitary prime factorization of H_d is *unique* in the sense of Definition 2 given below. On the other hand, the language H^* considered above has many infinitary prime factorizations. For instance, any infinite product of factors $(\varepsilon + w)$, where w runs through all nonempty words of H^*, constitutes such a factorization. Thus, the language H^* possesses both a prime factorization and (nondenumerably) many infinitary prime factorizations.

Some clarifying remarks are in order. When we consider infinite products $\prod_{i=1}^{\infty} L_i$, where each L_i is a language, we consider only finite words defined by the product. Then we also assume that each L_i contains the empty word. Indeed, an infinite product of languages defines finite words only if all of these languages, with at most finitely many exceptions, contain the empty word. In this case there is a language K and an integer $m \geq 1$ such that the original product can be written as

$$\prod_{i=1}^{\infty} L_i = K\prod_{i=m}^{\infty} L_i,$$

where each language in the product on the right side contains the empty word. If every language in the product $\prod_{i=1}^{\infty} L_i$ contains the empty word, then each word in the product belongs to a finite prefix of the product.

In the following definition we assume that each of the languages L_i and K_i *properly* contains the empty word.

Definition 2. A language L has a *unique infinitary prime factorization* if $L = \prod_{i=1}^{\infty} L_i$, where each L_i is prime and, whenever $L = \prod_{i=1}^{\infty} K_i$, where each K_i is prime, then $L_i = K_i$, for all i. If L is over a one-letter alphabet, it is only required that the languages K_i are the languages L_i in some order.

Since languages over one letter are commutative, the relaxation of uniqueness given in the definition is very natural. It is not difficult to see that the infinitary prime factorization given above for the language H_d is unique. A language can have both a prime factorization in the sense of Definition 1 and an infinitary prime factorization. For instance, as seen above, Σ^* has a prime factorization and also an infinitary prime factorization

$$\Sigma^* = \prod_{w}(\varepsilon + w),$$

where w runs through all nonempty words over Σ.

3. Comparisons between different notions

We now summarize the many notions of prime factorizations discussed above. Thus, a prime factorization (or decomposition) of a language can be

1. finitary,
2. unique finitary,
3. strong finitary,
4. infinitary,
5. unique infinitary.

Altogether this gives numerous possibilities for a language L: each of the properties (1)–(5) can be present or absent. Many combinations are excluded by definition. For instance, if L has property (2) (resp., 5), it surely possesses also property (1) (resp., 4). For most of the 32 subsets of the properties (1)–(5), it is easy to give an example of a language possessing each of the properties in that subset but none of the others, or show that no such language exists. We now present a few examples, some of them from [5].

The languages L_1, L_2, and L_3 defined below are over the one-letter alphabet $\{a\}$. Thus, for them uniqueness is up to the order of the factors.

The language L_1 consists of all words a^n such that every number 1 in the binary representation of n occurs in an even position, counted from the right. Hence, the six shortest words in L are

$$\varepsilon,\ a^2,\ a^8,\ a^{10},\ a^{32},\ a^{34}.$$

The sets $\{\varepsilon, a^{2^{2n+1}}\}$, $n \geq 0$, constitute the collection of prime languages appearing in any decomposition of L_1. This means that L_1 has no finitary but a unique infinitary prime factorization. These properties are also shared by the language L_2 defined as follows. It consists of all words a^i such that the binary representation of i is in the regular language

$$(00000 + 00001 + 00010 + 00100 + 01000)^*.$$

But now the factors in the prime factorization are of cardinality 5, instead of the cardinality 2 in L_1.

The language L_3 consists of all words a^i such that the binary representation of i is in the regular language

$$(000 + 010 + 100 + 110 + 011 + 101 + 111)^*.$$

Then L_3 has no finitary prime factorization but nondenumerably many infinitary ones.

Clearly, the language $L_i b$, $1 \leq i \leq 3$ has no prime factorization at all in any of our five senses.

We already noticed that Σ^* has both a finitary and an infinitary prime factorization, in fact, infinitely many of both. The following is a basic open question.

Open problem: Can a language have both a finitary prime factorization and a *unique* infinitary one?

4. Length codes

We now introduce a notion useful in considerations about prime factorizations of finite languages. We believe that the notion is also important on its own right.

Definition 3. A language L is a *length code* if every equation

$$u_1 \cdots u_k = v_1 \cdots v_l, \quad u_i, v_j \in L, \quad 1 \leq i \leq k, \quad 1 \leq j \leq l,$$

implies that $k = l$.

Clearly, every *code* (see [12]) is also a length code. The converse does not hold true. For instance, the language $\{a, ab, ba\}$ is not a code but it is a length code. This can be seen as follows.

Consider an equation as in Definition 3. We may assume that

$$u_1 = a, \quad v_1 = ab$$

because, otherwise, the equation can be shortened or is not valid. Consequently, $u_2 = ba$, which implies that $v_2 = a$ (leads to a shortening) or $v_2 = ab$ (leads to a loop). Therefore, the only possibility is the equation

$$(ab)^i a = a(ba)^i, \quad i \geq 0.$$

The notions of a code and a length code can very naturally be defined in terms of morphisms. Apart from length codes, this approach also leads to the notion of a *Parikh code*. By definition, a morphism $h : \Sigma^* \to \Delta^*$ is a *code* (resp., a *Parikh code, a length code*) if the equation $h(x) = h(y)$ always implies the equation $x = y$ (resp., $\Psi(x) = \Psi(y)$, $|x| = |y|$). (Here $\Psi(x)$ denotes the Parikh vector of x.) It follows that the set of codes is included in the set of Parikh codes which, in turn, is included

in the set of length codes. That the latter inclusion is proper is a consequence of the example above: the length code given is not a Parikh code. We will see below that also the former inclusion is proper.

The notions of a length code and a Parikh code have appeared earlier in the literature under different names. Length codes were called *numerically decipherable* in [14] and *precodes* in [8]. As far as we know, Parikh codes were introduced in [10]. There they appear under the name of *multiset decipherable codes*.

Following the ideas in [10], we now construct a Parikh code that is not a code. Consider the morphism

$$h : \{\alpha, \beta, \gamma, \delta\}^* \to \{a, b\}^*$$

defined by

$$h(\alpha) = bba, \quad h(\beta) = bbabb, \quad h(\gamma) = bab, \quad h(\delta) = abbbabab.$$

Since $h(\alpha\beta\gamma\delta) = h(\beta\delta\alpha\gamma)$, we conclude that h is not a code. A straightforward case analysis shows that h is a Parikh code. Indeed, from $h(x) = h(y)$, $x \neq y$, we conclude first that α (resp., β) is a prefix of x (resp., y). The possible continuations of x are now α and β, of which the former leads to an immediate contradiction. Proceeding in this way we reach the result

$$x = \alpha\beta\gamma\delta, \quad y = \beta\delta\alpha\gamma,$$

which is, consequently, the only possibility.

Sets $H \subseteq \Sigma^n$ considered in Section 2 are codes and, hence, also length codes. We are now ready to generalize the prime decomposition of H^* (considered above) to concern all regular length codes.

Theorem 2. *If L is a regular length code, then L^* has a prime decomposition consisting of two regular factors.*

Proof. It follows by the assumption that the empty word is not in L. Clearly,

$$L^* = (\varepsilon + L)\left(\varepsilon + L(L^2)^*\right).$$

We prove first that the second factor is prime. This follows because its decompositions must have the form

$$\varepsilon + L(L^2)^* = (\varepsilon + H_1)(\varepsilon + H_2), \quad H_1, H_2 \subseteq L(L^2)^*,$$

and, consequently, each word in $H_1 \cup H_2$ is a product of an odd number of words in L. Since L is a length code, any catenation of a word in H_1 and a word in H_2 is a product of an even number of words in L, which is impossible. Hence, one of the sets H_1 and H_2 is empty and, consequently, the left side is prime.

To conclude the proof, we show that show that $\varepsilon + L$ is prime. Assume that we can write $\varepsilon + L = (\varepsilon + K_1)(\varepsilon + K_2)$, $\varepsilon \notin K_i$, $i = 1, 2$. It follows that $K_i \subseteq L$, $i = 1, 2$. This means that there exist $u_i \in K_i \subseteq L$, $i = 1, 2$ such that $u_1u_2 \in L$. This contradicts the fact that L is a length code. □

Observe that a language L is not a length code exactly in case, for some i and j, $i \neq j$, we have $L^i \cap L^j \neq \emptyset$. In general, the minimal difference between i and j can be arbitrarily large. For any $t \geq 2$, there is a finite language F_t that is not a length code but $F_t^i \cap F_t^j = \emptyset$ wherever $1 \leq |i - j| < t$. For instance, this holds for

$$F_t = \left\{(ab)^{3t}, aba, bab\right\}.$$

We now consider the problem of how long words we have to test in order to find out that a finite language F is a length code. We already noticed that there is no upper bound, independent of F, for the minimal difference between the number of factors i and j in two decompositions. On the other hand, there is an upper bound for both i and j depending on F (in the sense made precise below).

By a *proper suffix* of a word w we mean a suffix of w different from w and the empty word. By m_F we denote the length of the longest word(s) in F, by S_F the set of all proper suffixes of the words in F, and s_F the cardinality of S_F. (Thus, a suffix appearing in several words of F is counted only once.) Finally, we denote $c_F = m_F s_F$.

Theorem 3. *Assume that F is a finite language (not containing ε) such that the inequality $F^i \cap F^j \neq \emptyset$ holds for some i and j, $i \neq j$. Then, for some i_1 and j_1, $i_1 \neq j_1$,*

$$F^{i_1} \cap F^{j_1} \neq \emptyset, \quad i_1 \leq 2c_F, \quad j_1 \leq 2c_F.$$

Proof. By the assumption, we have

$$u_1 \cdots u_i = v_1 \cdots v_j, \ i \neq j, \qquad (*)$$

where each of the u-words and v-words is in F. We assume that this equation is *minimal* with respect to i and j, that is, there are no i' and j', $i' \neq j'$, $i' < i$, $j' < j$, such that $x_1 \cdots x_{i'} = y_1 \cdots y_{j'}$, where each of the x-words and y-words is in F. To

complete the proof we have to show that, under this minimality assumption, neither one of the indices i and j exceeds $2c_F$. Indeed, we are going to prove that if one of the indices i and j exceeds $2c_F$, then we can remove a nonempty factor from both sides of the equation.

If $u_1 = v_1$ then $u_2 \cdots u_i = v_2 \cdots v_j$, which contradicts the minimality assumption. Thus, one of the words is a proper prefix of the other, say, $v_1 = u_1 s_1$, $s_1 \neq \varepsilon$. We say that the suffix $s_1 \in S_F$ *appears*. We now read u-words until we obtain the first one, say u_α, such that $|u_1 \cdots u_\alpha| > |v_1|$. Then we read v-words until we obtain the first one, say v_β, such that $|v_1 \cdots v_\beta| > |u_1 \cdots u_\alpha|$. Consequently,

$$v_1 \cdots v_\beta = u_1 \cdots u_\alpha s_2, \quad s_2 \in S_F,$$

and we say that the suffix s_2 *appears*. Observe that $\alpha, \beta \leq m_F$. The process is continued. We always read u-words until the part read exceeds in length the v-part so far read, and then again v-words until the u-part is exceeded in length. In this way we get also a sequence $s_1, s_2, \ldots$ of appearing suffixes, where for all $i \geq 1$, if

$$v_1 \cdots v_{\beta'} = u_1 \cdots u_{\alpha'} s_{i+1},$$

then $\alpha', \beta' \leq im_F$.

Assume now that one of original indices i and j exceeds $2c_F$. We will show that this contradicts minimality.

The assumption implies that we reach a situation

$$v_1 \cdots v_{\beta''} = u_1 \cdots u_{\alpha''} s_{j+1},$$

for some j, where either $\beta'' > 2s_F m_F$ or $\alpha'' > 2s_F m_F$. On the other hand, $\alpha'', \beta'' \leq jm_F$. This is possible only if $j > 2s_F$. Therefore, some suffix s' appears at least three times.

Consequently, there are indices

$$\alpha_1 < \alpha_2 < \alpha_3 \quad \text{and} \quad \beta_1 < \beta_2 < \beta_3$$

such that

$$u_1 \cdots u_{\alpha_t} s' = v_1 \cdots v_{\beta_t}, \quad 1 \leq t \leq 3.$$

This means that if we remove the factor $u_{\alpha_1+1} \cdots u_{\alpha_2}$ from the left and the factor $v_{\beta_1+1} \cdots v_{\beta_2}$ from the right side of the equation $(*)$, a valid equation still results. However, if

$$(\alpha_2 - \alpha_1) - (\beta_2 - \beta_1) = i - j,$$

the number of u-factors in the new equation equals the number of v-factors, and we do not get a counterexample. Assume this is the case.

We can also remove the factor $u_{\alpha_2+1} \cdots u_{\alpha_3}$ from the left and the factor $v_{\beta_2+1} \cdots v_{\beta_3}$ from the right side of the equation $(*)$, without affecting the validity of the equation. If $\alpha_3 - \alpha_2 = \beta_3 - \beta_2$, we perform only this removal, and not the one discussed in the preceding paragraph. Otherwise, we perform both removals. In every case the new equation contradicts minimality and has a different number of u- and v-factors on its two sides. □

The argument is simpler and the bound $m_F s_F$ is sufficient if we are dealing with codes. By an *F-factorization* of a word w we mean an equation $w = u_1 \cdots u_i$, where each of the words on the right side is in F. The proof of the following result follows the lines of the preceding proof.

Theorem 4. *Assume that a finite language F not containing the empty word is not a code. Then some word in $F^{m_F s_F}$ has two different F-factorizations.*

Theorem 3 and its proof can also be used to construct an algorithm for deciding whether or not a given finite language is a length code. This leads to a definition of a *domino graph*, similarly as in [14]. We omit the details. In our case the time complexity will be of the order $c_F \# F$, where $\# F$ is the cardinality of F.

Length codes play an important role when dealing with prime decompositions of languages. Languages over the unary alphabet of cardinality at least two are not length codes. The language

$$F_1 = K = \{ab, aba, bab\}$$

constitutes a simple example over the binary alphabet. Indeed,

$$ababab = (ab)(ab)(ab) = (aba)(bab)$$

and, thus, the same word equals the product of both two and three words of K.

In the sequel we will be quite much concerned with languages such as K^*. We will see that K^* possesses a prime decomposition of two factors but a technique very different from the one in Theorem 2 is needed.

5. Techniques for star languages

We now consider some special techniques for obtaining prime decompositions for star languages. It turns out that the resulting factors are sometimes "strange" in comparison with the original language. We begin with the following simple result.

Theorem 5. *If a language L is prime then, for every nonempty word $w \in L$, there is a word $w' \in L$ (resp., $w'' \in L$) such that $ww' \notin L$ (resp., $w''w \notin L$).*

Proof. Assume the contrary: no such word w' exists for a nonempty word $w \in L$. This means that $ww' \in L$, for all words $w' \in L$. Consequently, $(\varepsilon + w)L = L$, which shows that L is not prime. If no w'' exists, we obtain similarly $L(\varepsilon + w) = L$. □

The *converse* of Theorem 5 is not valid. For instance, the language

$$L = \varepsilon + \left\{ab^{2i+2}a, ab^{2j+1}a, ab^{2i+2}a^2b^{2j+1}a | i, j \geq 0\right\}$$

satisfies the condition of Theorem 5. Indeed, for every nonempty $w \in L$, we have $w^2 \notin L$. However, L is not prime because

$$L = \left(\varepsilon + \left\{ab^{2i+2}a | i \geq 0\right\}\right)\left(\varepsilon + \left\{ab^{2j+1}a | j \geq 0\right\}\right).$$

We will now establish a general result about prime decompositions of star languages. The language K^* considered above will have a prime decomposition consisting of two languages.

Let F be a finite language not containing the empty word. We say that a word $x \in F$ is *independent* if no other word of F appears as a prefix of x, and x itself is not a prefix of any other word of F. Clearly, if a longest (resp., shortest) word in F has no proper prefix in F (resp., is not a proper prefix of any word in F), then it is independent.

Theorem 6. *If F is a finite language containing an independent word, then F^* has a prime decomposition consisting of two factors.*

Proof. Let m be the greatest word length in F. (In the estimates below we assume that the smallest word length in F equals 1. The estimates will be better if the smallest word length is greater, but this is irrelevant for our purposes.) We define

$$H_1 = \bigcup_{i_1} F^{i_1},$$

where i_1 runs through all integers ≥ 1 except the integers in the *open* intervals

$$\left((2m+1)^{4t}, (2m+1)^{4t+1}\right) \quad \text{and} \quad \left((2m+1)^{4t+1}, (2m+1)^{4t+2}\right), \quad t \geq 1.$$

Thus $i_1 \geq 1$ misses exactly the integers i satisfying

$$(2m+1)^{4t} < i < (2m+1)^{4t+1} \quad \text{or} \quad (2m+1)^{4t+1} < i < (2m+1)^{4t+2},$$

for some $t \geq 1$. Similarly, we define

$$H_2 = \bigcup_{i_2} F^{i_2},$$

where i_2 runs through all integers ≥ 1 except the integers in the open intervals

$$\left((2m+1)^{4t+2}, (2m+1)^{4t+3}\right) \quad \text{and} \quad \left((2m+1)^{4t+3}, (2m+1)^{4t+4}\right), \quad t \geq 1.$$

We claim that

$$F^* = (\varepsilon + H_1)(\varepsilon + H_2)$$

is a prime decomposition of F^*.

Observe first that the equation is valid because $H_1 \cup H_2 = F^+$. Indeed, i_1 and i_2 miss two disjoint sets of integers. We still have to prove that the two factors on the right side are prime. It suffices to carry out the proof for $\varepsilon + H_1$ because the proof for $\varepsilon + H_2$ is essentially the same.

Any factorization of $\varepsilon + H_1$ can be written in the form

$$\varepsilon + H_1 = (\varepsilon + N_1)(\varepsilon + N_2), \quad \varepsilon \notin N_1 \cup N_2.$$

The idea is to prove that one of the sets N_1 and N_2 must be empty.

We fix the value of t, and consider the set

$$F^\alpha, \quad \alpha = (2m+1)^{4t+1}.$$

Thus, $F^\alpha \subseteq H_1$ but elements in the sets F^i, where i is close to α, are in H_1 only in case they also belong to some set F^j, where j is in the range of i_1.

Let C_1 be the subset of N_1 (possibly empty) consisting of elements of F^α. Let B_1 (B for "below") be the subset of N_1 consisting of elements of all sets F^j, where $j < \alpha$ is in the range of i_1. Let A_1 (A for "above") be the subset of N_1 consisting of elements of all sets F^j, where $j > \alpha$ is in the range of i_1. Let subsets C_2, B_2, A_2 of N_2 be similarly defined.

The shortest word in C_1 is of length $\geq (2m+1)^{4t+1}$. The longest word in B_1 is of length $\leq m(2m+1)^{4t}$. Thus, the shortest word in C_1 is longer than twice the length of the longest word in B_1. Similarly, the shortest word in A_1 is longer than twice the length of the longest word in C_1. Consequently, the sets C_1, B_1, A_1 are pairwise mutually disjoint. The same considerations apply to the sets C_2, B_2, A_2. Observe that

$$N_1N_2 = (A_1 \cup B_1 \cup C_1)(A_2 \cup B_2 \cup C_2).$$

Assume first that one of the sets C_1 and C_2, say C_1, is empty. Let w be a word in C_2 of maximal length $m\alpha$. If B_1 is not empty, N_1N_2 contains a word of length

$$j + m\alpha, \quad 1 \leq j \leq m(2m+1)^{4t}.$$

But no such word is in H_1 and, hence, B_1 must be empty. If C_2 is empty, we conclude similarly that B_2 is empty.

Assume, secondly, that both of the sets C_1 and C_2 are nonempty. Choose arbitrary words $w_i \in C_i$, $i = 1, 2$. Thus $w_1, w_2 \in F^\alpha$. The word w_1w_2 is in N_1N_2 and, therefore, in H_1. This means that it belongs to one of the powers F^{i_1} defining H_1. Since

$$2\alpha \leq |w_1w_2| \leq 2m\alpha,$$

the only such power is F^α. We conclude that the word w_1w_2 can be expressed as a catenation of 2α words in F, as well as a catenation of α words in F. This is impossible if w_1 equals x^α, where $x \in F$ is independent. Thus, x^α is not in C_1 or C_2 but it must be in H_1. It cannot be in C_1C_2 because x^α is not a catenation of 2α words in F. But it can also not be in B_1C_2 or any other product defining N_1N_2. Thus, altogether the second case is impossible and, consequently, one of the sets C_1 and C_2 is empty. As shown above, also the corresponding B-set is empty.

By choosing t large enough, we can get any specific words into the B-sets. Therefore, one of the factors in our factorization of $\varepsilon + H_1$ is trivial, which concludes the proof of Theorem 6. □

The language K considered above contains the independent word *bab*. Thus, K^* has a prime decomposition into two factors.

It seems likely that Theorem 6 remains valid without the assumption concerning the independent word. The simultaneous parsing of a word into 2α and α words in F seems always to lead into a contradiction.

If we consider arbitrary (finite) prime decompositions, instead of ones with only two factors, we obtain the result without the assumption concerning the independent word.

Theorem 7. *If F is finite, then the language F^* has a prime decomposition.*

Proof. We follow the notations in the preceding proof and assume that a decomposition

$$\varepsilon + H_1 = (\varepsilon + N_1)(\varepsilon + N_2), \quad \varepsilon \notin N_1 \cup N_2,$$

where both N_1 and N_2 are nonempty, is possible. It follows that, for some t and the corresponding α, both the the sets C_1 and C_2 are nonempty. We choose the smallest such $t = t_0$ and conclude that C_1 and C_2 are nonempty also for every $t_1 > t_0$. (Otherwise, we would get a contradiction, as in the preceding proof, by considering the B-sets corresponding to t_1.) Assume that we have a nontrivial decomposition

$$\varepsilon + N_1 = \left(\varepsilon + N_1^1\right)\left(\varepsilon + N_1^2\right), \quad \varepsilon \notin N_1^1 \cup N_1^2.$$

Hence,

$$\varepsilon + H_1 = \left(\varepsilon + N_1^1\right)\left(\varepsilon + N_1^2\right)(\varepsilon + N_2).$$

Considering the C-sets for $t = t_0$ in the same way as before, we conclude the existence of a word $w \in F^\alpha$ that also belongs to $F^{3\alpha}$. Since α is fixed, such decompositions cannot continue forever. This means that $\varepsilon + N_1$ has a (finitary) prime decomposition. This is true also for $\varepsilon + N_2$ and, hence, for $\varepsilon + H_1$. Thus, also $\varepsilon + H_2$ and F^* have finitary prime decompositions. □

We *conjecture* that every regular language has a prime decomposition. This result was established in [5] for regular languages over a one-letter alphabet. Theorem 7 constitutes a step for a possible proof in the general case. As such our constructions do not work for arbitrary star languages because then the length arguments fail.

6. On regular prime decompositions of K^* and related languages

We consider, finally, possible prime decompositions of K^* and related languages, that is, the case where K is not a length code. It seems very likely that, although K^* has a prime decomposition of two factors, it still does not have any *regular* prime decomposition (i.e., one where the factors are regular), not even an infinitary one. Thus, there would be regular languages having a prime decomposition but having no prime decomposition consisting of regular languages.

Any prime decomposition of K^* is of the form

$$K^* = (\varepsilon + H_1)(\varepsilon + H_2)\cdots(\varepsilon + H_n),$$

where the languages H_i, $1 \leq i \leq n$, are contained in K^* and do not contain the empty word. Moreover, at least one of them has to be infinite.

The most immediate decompositions for star languages (also in use in Theorem 2) are of the form

$$L^* = (\varepsilon + L + \ldots L^{m-1})(\varepsilon + L^m(L^m)^*), \quad m \geq 2.$$

However, the following result shows that no decomposition of this type can lead to a prime decomposition of K^*.

Lemma 1. *An infinite union*

$$K_1 = \varepsilon + \bigcup_{j=1}^{\infty} K^{i_j}, \quad 1 \leq i_1 < i_2 < \cdots,$$

is not prime, provided there is a bound B such that $i_{j+1} - i_j \leq B$, for all $j \geq 1$.

Proof. Choose the smallest r such that $i_r \geq 3B$, and consider the word

$$w = (ab)^{i_r} = (ab)^{3B}(ab)^{\nu},$$

where $\nu \geq 0$. Let $w' \in K_1$ be arbitrary. Clearly,

$$ww' \in K^{3B+\nu+i_s},$$

for some i_s. (If $w' = \varepsilon$, then $i_s = 0$.) Each of the B factors $(ab)(ab)(ab)$ constituting the prefix of w can be parsed also $(aba)(bab)$, which shows that

$$ww' \in K^{3B+\nu+i_s-\mu}, \quad \text{for all } \mu, \quad 0 \leq \mu \leq B.$$

By the assumption concerning B, there is a μ, $0 \leq \mu \leq B$, such that $3B + \nu + i_s - \mu$ is one of the exponents i_j. This shows that $ww' \in K_1$. Since $w' \in K_1$ was arbitrary, we conclude that

$$(\varepsilon + w)K_1 = K_1.$$

This shows that K_1 is not prime. □

As regards regular languages, the result of Lemma 1 can be presented in the following form.

Lemma 2. *No regular language $R \subseteq K^*$, containing the empty word and infinitely many powers of K, is prime.*

Proof. Since R is regular, there is a bound B between the exponents of consecutive powers of K in R, as in Lemma 1. In addition, R may contain "loose" words that do not belong to any full power of K contained in R. The argument in the proof of Lemma 1 remains valid, with K_1 replaced by R. □

According to Lemma 2, in any regular prime decomposition of K^*, finitary or infinitary, every factor contains only finitely many full powers of K.

Instead of K, we can start with any finite language L that is not a length code. (Since L does not necessarily contain an independent word, we do not get a prime decomposition of two factors but have to use Theorem 7.) We obtain, thus, the following summarizing result.

Theorem 8. *If L is a regular length code, then L^* has a prime decomposition consisting of two regular factors. If L is finite but not a length code, then in any regular prime decomposition of L^*, finitary or infinitary , every factor contains only finitely many full powers of L. However, L^* has a (finitary) prime decomposition.*

7. Conclusion

The notion of a length code is interesting and seems to be applicable in various contexts. We hope to return to a further study of it. Some of the basic problems concerning products and primality of languages are challenging. As we have seen, one of such problems deals with the prime decompositions of regular languages.

References

[1] S.V. Avgustinovich, A. Frid, A unique decomposition theorem for factorial languages, Int. J. Algebra Comput., 15 (2005) 149–160.
[2] J. Czyzowicz, W. Fraczak, A. Pelc, W. Rytter, Linear-time prime decompositions of regular prefix codes, Int. J. Found. Comput. Sci. 14 (2003) 1019–1031.
[3] M. Daley, M. Domaratzki, K. Salomaa, On the operational orthogonality of languages, in: M. Kunc, A. Okhotin (Eds.), Proceedings of the First International Workshop on Theory and Applications of Language Equations, Turku Centre for Computer Science General Publication No. 44, 2007, pp. 43–53.
[4] A. Frid, Commutation in binary factorial languages, in: Proceedings of DLT'2007, LNCS, vol. 4588, Springer, 2007, pp. 193–204.
[5] Y.-S. Han, A. Salomaa, K. Salomaa, D. Wood, S. Yu, Prime decompositions of regular languages, Theory Comput. Sci. 376 (2007) 60–69.
[6] Y.-S. Han, Y. Wang, D. Wood, Infix-free regular expressions and languages, Int. J. Found. Comput. Sci., in press.
[7] Y.-S. Han, D. Wood, The generalization of generalized automata: expression automata, in: Implementation and Application of Automata, CIAA'04, LNCS, vol. 3317, Springer, 2005, pp. 156–166.
[8] T. Head, A. Weber, Deciding code related properties by means of finite transducers, in: R. Capocelli, U. De Santis, U. Vaccaro (Eds.), Sequences, vol. II, Springer-Verlag, 1993, pp. 260–272.
[9] M. Kunc, The power of commuting with finite sets of words, Theory Comput. Syst. 40 (2007) 521–551.
[10] A. Lempel, On multiset decipherable codes, IEEE Trans. Inform. Theory IT-32 (1986) 714–716.
[11] A. Mateescu, A. Salomaa, S. Yu, Factorizations of languages and commutativity conditions, Acta Cybern. 15 (2002) 339–351.
[12] G. Rozenberg, A. Salomaa (Eds.), Handbook of Formal Languages, vol. 1–3, Springer-Verlag, 1997.
[13] A. Salomaa, S. Yu, On the decomposition of finite languages, Proceedings of Developments in Language Theory (DLT'99), World Scientific, 2000, pp. 22–31.
[14] A. Weber, T. Head, The finest homophonic partition and related code concepts, IEEE Trans. Inform. Theory IT-42 (1996) 1569–1575.